From Raw Data to Physics: Reconstruction and Analysis

Reconstruction: Tracking; Particle ID

How we try to tell particles apart

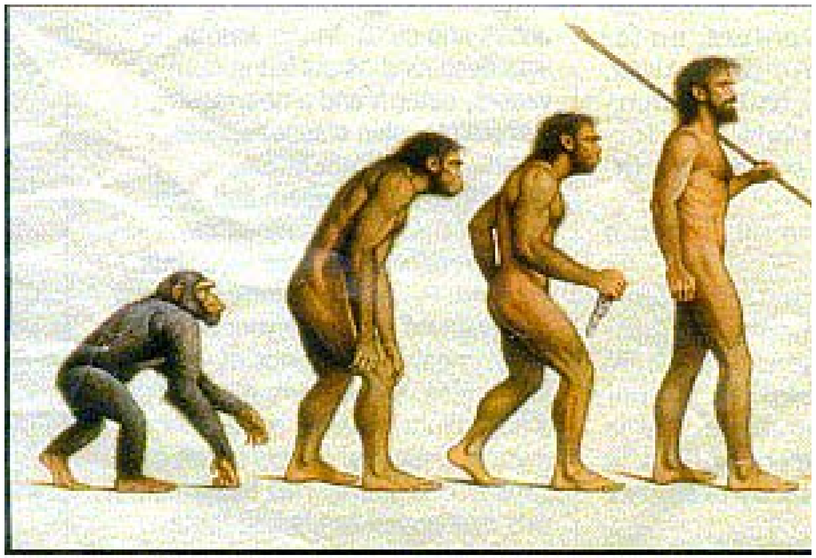
Analysis: Measuring α_S in QCD

What to do when theory doesn't make clear predictions

Alignment

We know what we designed; is it what we built?

Summary



From Raw Data to Physics:
Reconstruction and Analysis

Reconstruction: Particle ID

How we try to tell particles apart

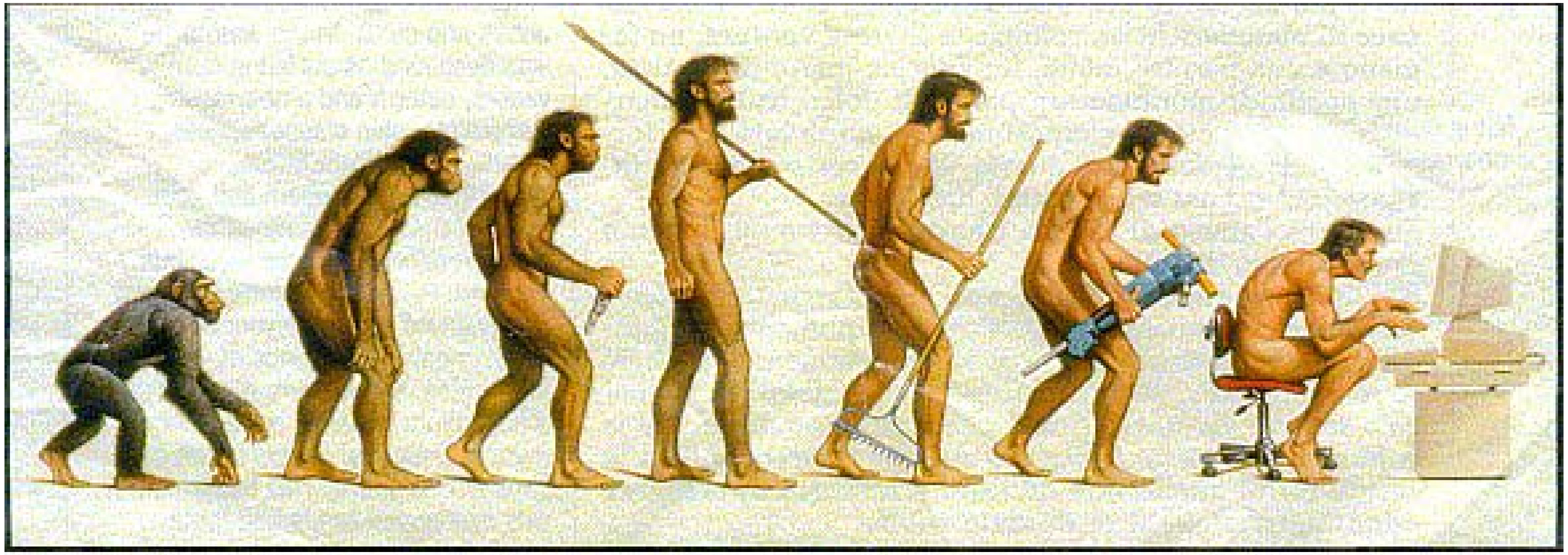
Analyzing simulated data: Measuring α_S in QCD

What to do when theory doesn't make clear predictions

Alignment:

We know what we designed; is it what we built?

Computing:



Somewhere, something went terribly wrong

Why does tracking need to be done well?

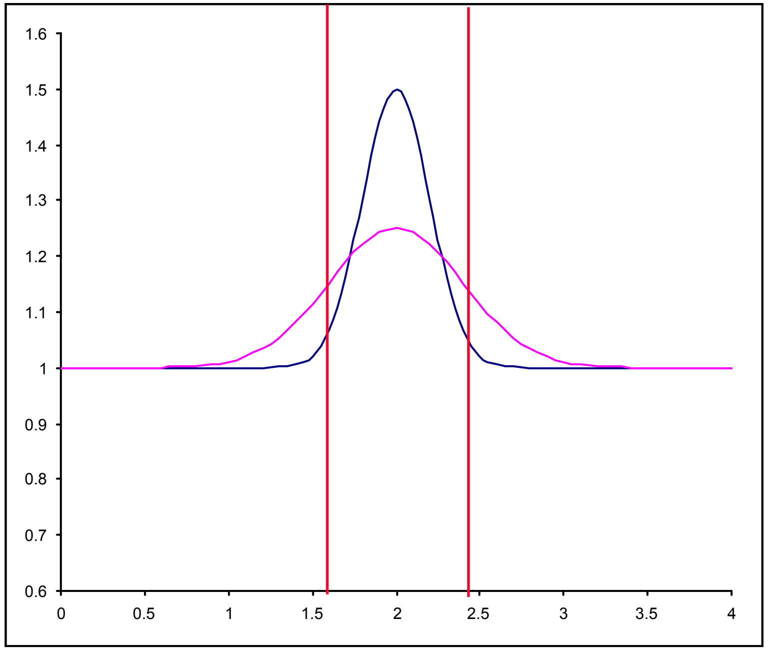
1) Tells you particles were created in an event

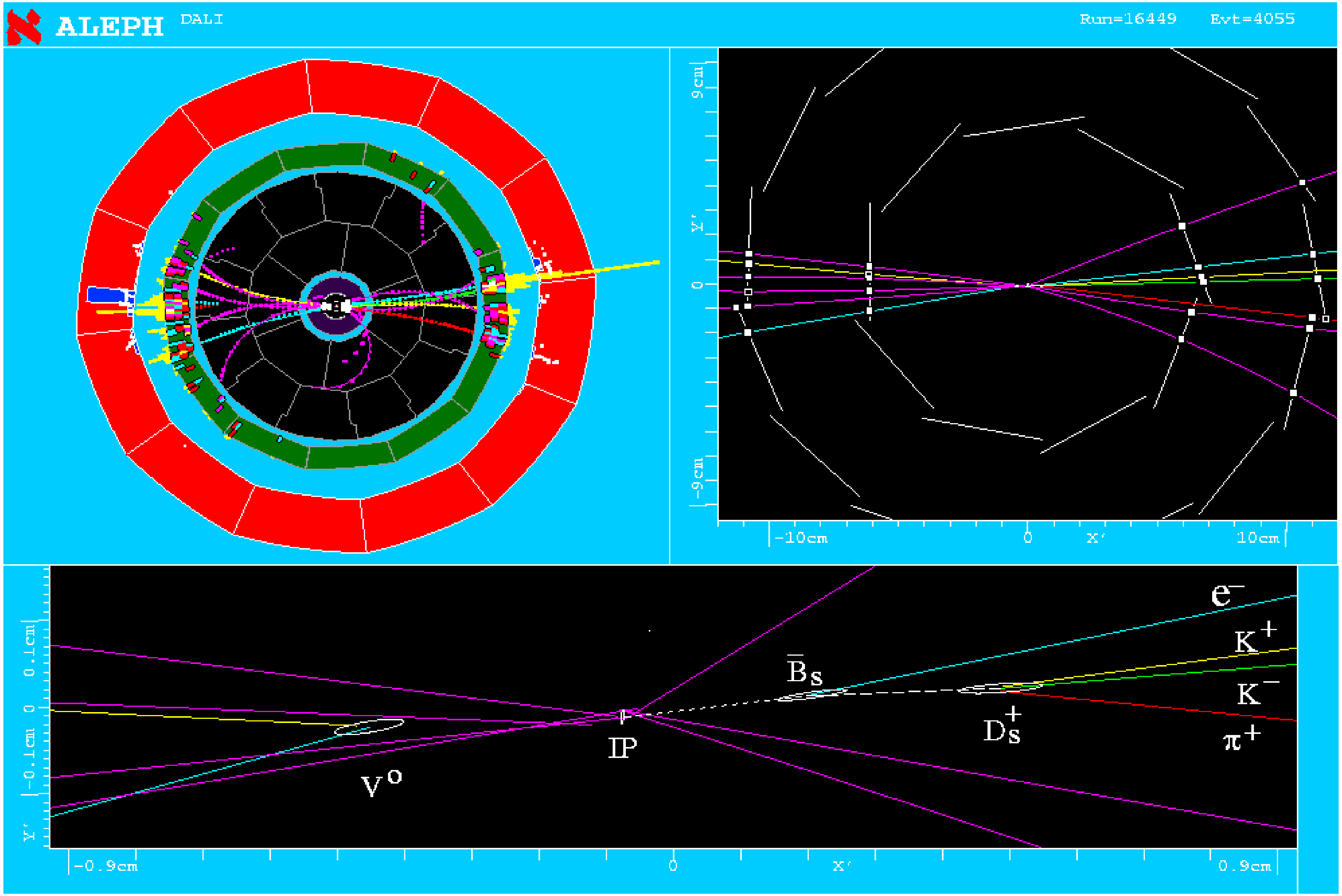
2) Allows you to measure their momentum

- Direction and magnitude
- Combine these to look for decays with known masses
- Only final particles are visible!

3) Allows you to measure spatial trajectories

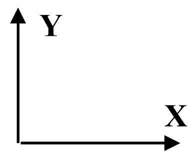
- Combine to look for separated vertices, indicating particles with long lifetimes





ALEPH
DALI
Run=16449
Evt=4055
9cm
Y'
0
-9cm
-10cm
0
X'
10cm
0.1cm
0
-0.1cm
Y'
-0.9cm
0
X'
0.9cm
e−
K+
K−
π+
$\bar{B}_s$
D_s^+
IP
V0

Track Fitting



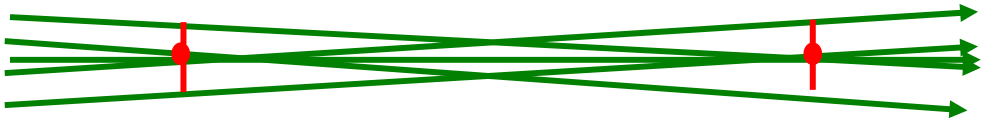
1D straight line as simple case

Two perfect measurements

- Away from interaction point
- With no measurement uncertainty
- Just draw a line through them and extrapolate

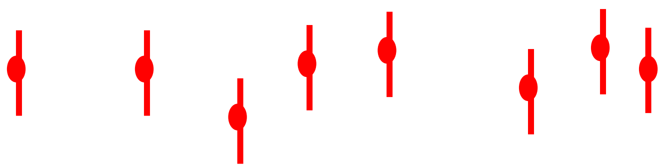
Imperfect measurements give less precise results

- The farther you go, the less you know



Smaller errors, more points help constrain the possibilities

How to find the best track from a large set of points?



How to fit quantitatively?

Parameterize track: $y(x) = \theta x + d$

- Two measurements, two parameters => OK

Best track?

- Consistency with measurements represented by $\chi^2 = \sum_{i=1}^{n_{hits}} \frac{\left(y_i - y(x_i)\right)^2}{\sigma_i^2}$

  Sum of normalized errors squared

Position of i^th hit — y_i

Predicted track position at i^th hit — $y(x_i)$

Accuracy of measurement — σ_i

- This is directly a function of our parameters:

$$\chi^2 = \sum_{i=1}^{n_{hits}} \frac{\left(y_i - \theta x_i - d\right)^2}{\sigma_i^2}$$

- The best track has the smallest normalized error
- So minimize in the usual way:

$$\frac{\partial \chi^2}{\partial \theta} = 0 \qquad \frac{\partial \chi^2}{\partial d} = 0$$

$$\frac{\partial \chi^2}{\partial \theta} = 2\sum \frac{(y_i - \theta x_i - d)}{\sigma_i^2}(-x_i)$$

$$0 = \left(\sum \frac{y_i x_i}{\sigma_i^2}\right) - \left(\sum \frac{x_i}{\sigma_i^2}\right) d - \left(\sum \frac{x_i^2}{\sigma_i^2}\right)\theta$$

$$\frac{\partial \chi^2}{\partial d} = 2\sum \frac{(y_i - \theta x_i - d)}{\sigma_i^2}(-1)$$

$$0 = \left(\sum \frac{y_i}{\sigma_i^2}\right) - \left(\sum \frac{1}{\sigma_i^2}\right) d - \left(\sum \frac{x_i}{\sigma_i^2}\right)\theta$$

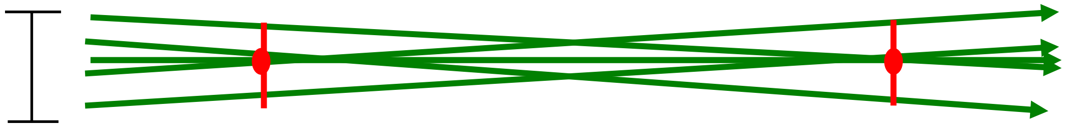
<u>Two</u> equations in <u>two</u> unknowns

- Terms in () are constants calculated from measurement, detector geometry

Generalizes nicely to 3D, helical tracks with 5 parameters

- Five equations in five unknowns

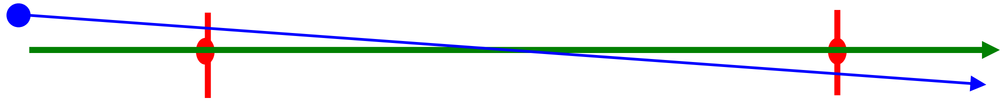
With a little more work, can calculate expected errors on θ, d



“Most likely” that <u>real</u> d (Y intercept) is within this band of $\pm\sigma_d$

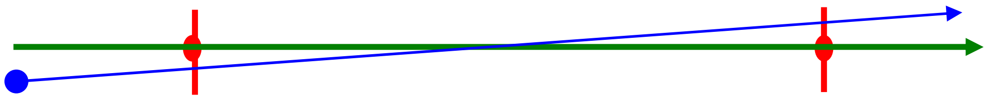
Similar θ error, where θ_{real} is most likely within $\pm\sigma_\theta$ of best value

Note that the errors are <u>correlated</u>:



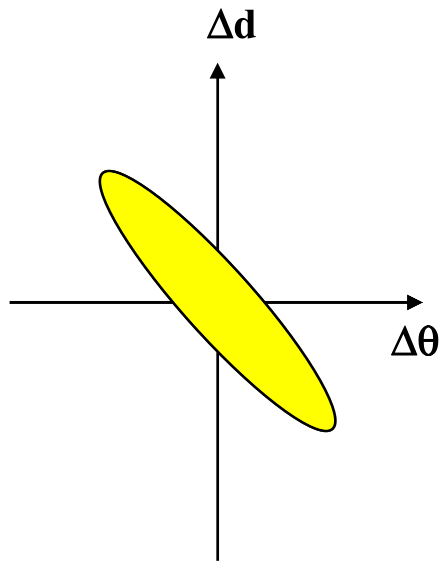
Δd = “+” - 0 > 0

$\Delta\theta$ = “-” - 0 < 0

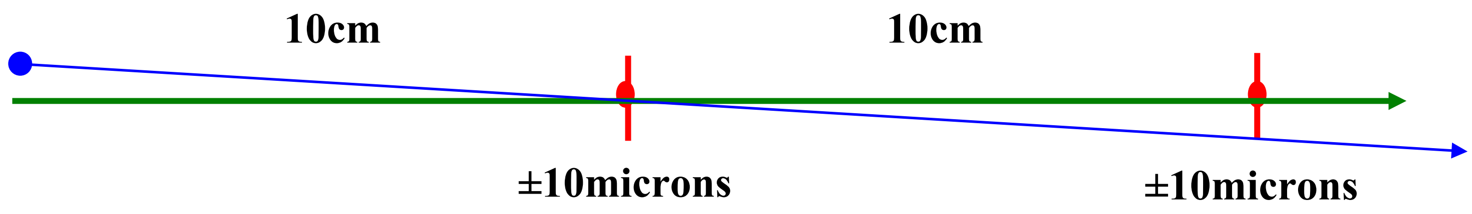


Δd = “-” - 0 < 0

$\Delta\theta$ = “+” - 0 > 0



Typical size of errors



Error on position is about ±10 microns

By similar triangles

Error on angle is about ±0.1 milliradians (±0.002 degrees)

Satisfyingly small errors!

Allows separation of tracks that come from different particle decays

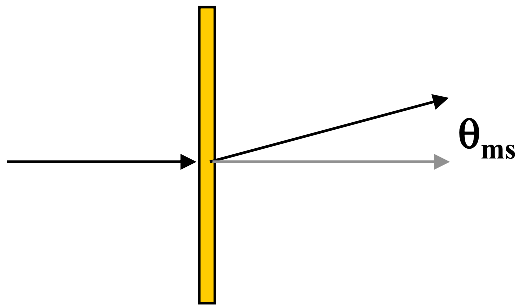
But how to we "see" particles?

- Charged particles pass through matter,
- ionize some atoms, leaving energy
- which we can sense electronically.

More ionization => more signal => more precision

=> more energy loss

Multiple Scattering

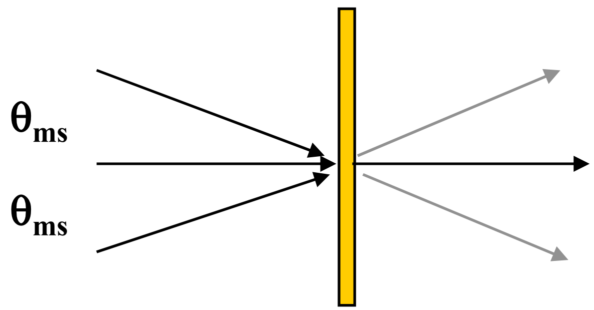


Charged particles passing through matter "scatter" by a random angle

$$\sqrt{\langle \theta_{ms}^2 \rangle} = \frac{15\,MeV/c}{\beta p}\sqrt{\frac{\text{thickness}}{X_{rad}}}$$

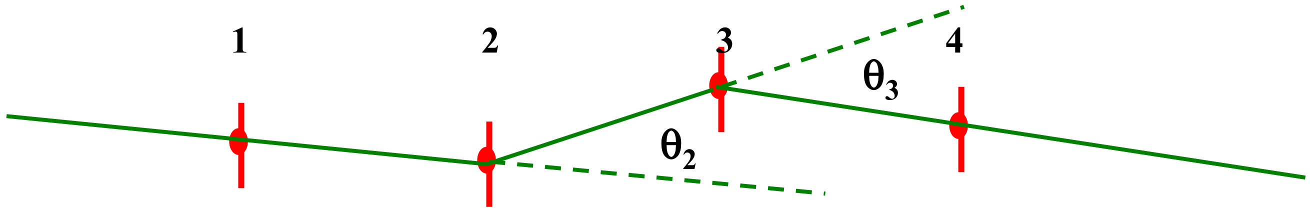
300μ Si RMS = 0.9 milliradians / βp

1mm Be RMS = 0.8 milliradians / βp



Also leads to position errors

So?



Fitting points 3 & 4 no longer measures angle at IP

Track already scattered by random angles θ_1, θ_2,θ_3

Track has more parameters

$$y(x) = d + \theta x + \theta_1 (x - x_1)\Theta(x - x_1)$$
$$+\theta_2 (x - x_2)\Theta(x - x_2) + \theta_3 (x - x_3)\Theta(x - x_3) + \ldots$$

1 if $x-x_3 > 0$, otherwise 0

If we knew θ_1, $\theta_{2,\ldots}$ we'd know entire trajectory

Can we measure those angles?

θ_2 roughly given by y_1, y_2, y_3

Just a more complex χ^2 equation?

$\sqrt{\langle \theta_{ms}^2 \rangle}$ **acts like a measurement**

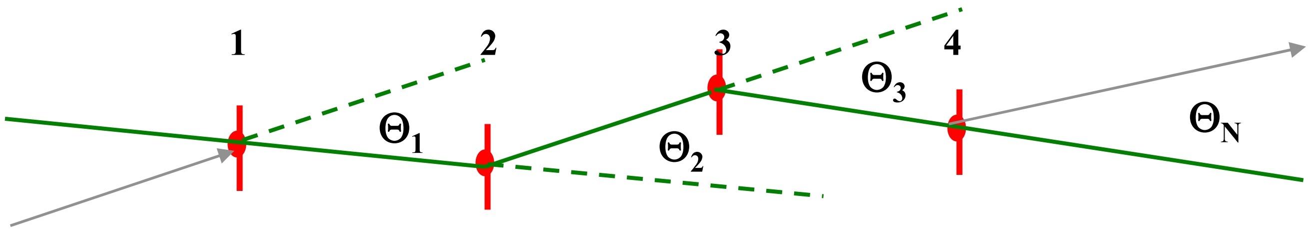
"I'd be surprised if it was larger than $0 \pm \frac{15MeV/c}{\beta p}\sqrt{\frac{L}{X_{rad}}}$

"Add information" to fit by adding new terms to χ^2

$$\chi^2 = \chi^2_{old} + \sum_i \frac{\theta_i^2}{\sigma_{ms}^2}$$

N measurements from planes (say 100)

N+2 unknowns (d, θ, plus N scattering angles)



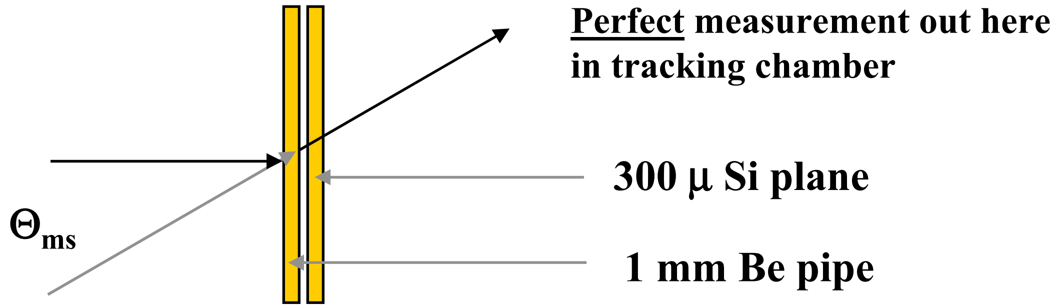
Can't see first, last scattering angles; can only extrapolate outside

Hence ignore θ_1, θ_N

Now all we have to do is solve 100 equations in 100 unknowns...

Nobody cares about θ_N

But θ_1 effects accuracy of d



θ_{ms} => 1.2 milliradian/βp error on θ

@10 cm, leads to 120μ/βp error on d

$$\sigma_d \approx 10\mu \oplus \frac{120\mu}{\beta p}$$

In spite of

N=100 chambers,

complicated programs

and inverting 100x100 matrices

Some problems, the programs can't fix!

“Kalman fit”?

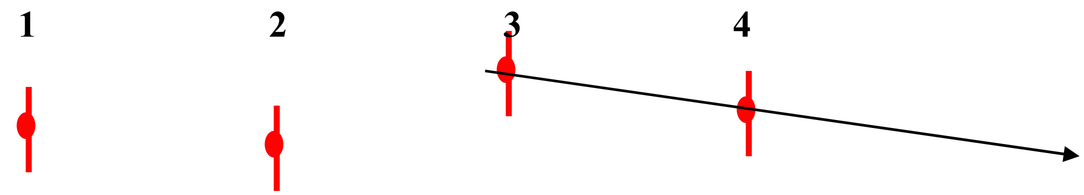
(ref: Brillion)

Computational expensive to calculate solutions with 100 angles

Computer time grows like $O(N^3)$, with N large

And we’re not really interested in all those angles anyway

Instead, approximate, working inward N times:



“Kalman fit”?

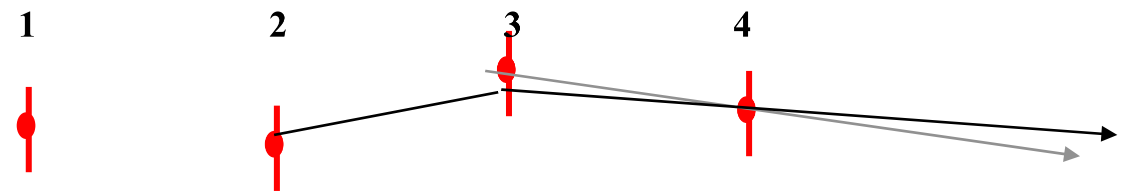
(ref: Brillion)

Computational expensive to calculate solutions with 100 angles

Computer time grows like $O(N^3)$, with N large

And we’re not really interested in all those angles anyway

Instead, approximate, working inward N times:



“Kalman fit”?

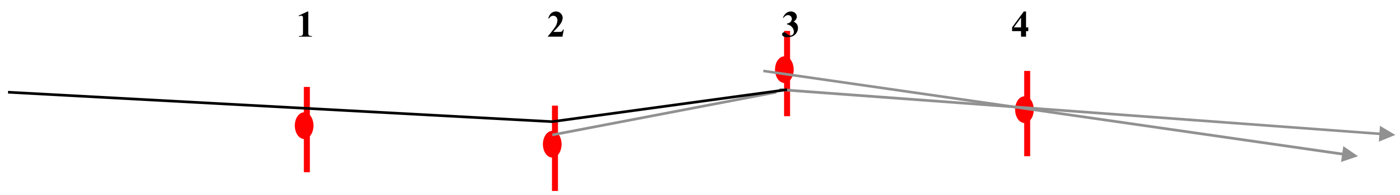
(ref: Brillion)

Computational expensive to calculate solutions with 100 angles

- Computer time grows like $O(N^3)$, with N large

And we’re not really interested in all those angles anyway

Instead, approximate, working inward N times:



This is O(N) computations

- May need to repeat once or twice to use good starting estimate
- Each one a little more complex
- But still results in a large net savings of CPU time

Moral: Consider what you <u>really</u> want to know

Particle ID (PID)

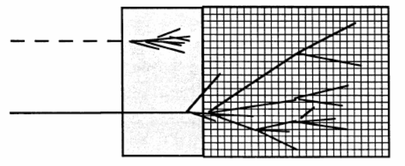
Track could be e, μ, π, K, or p; knowing which improves analysis

- Vital for measuring B->Kπ vs B->ππ rates
- Mistaking a π for e, μ, K or p increases combinatoric background

Leptons have unique interactions with material

- e deposits energy quickly, so expect E=p in calorimeter
- μ deposits energy slowly, so expect penetrating trajectory

But hadronic showers from π, K, p all look alike



Can't you measure mass from $m^2=E^2-p^2$?

For p=2GeV/c, pion energy = 2.005 GeV, kaon energy = 2.060 GeV

Calorimeters are not that accurate

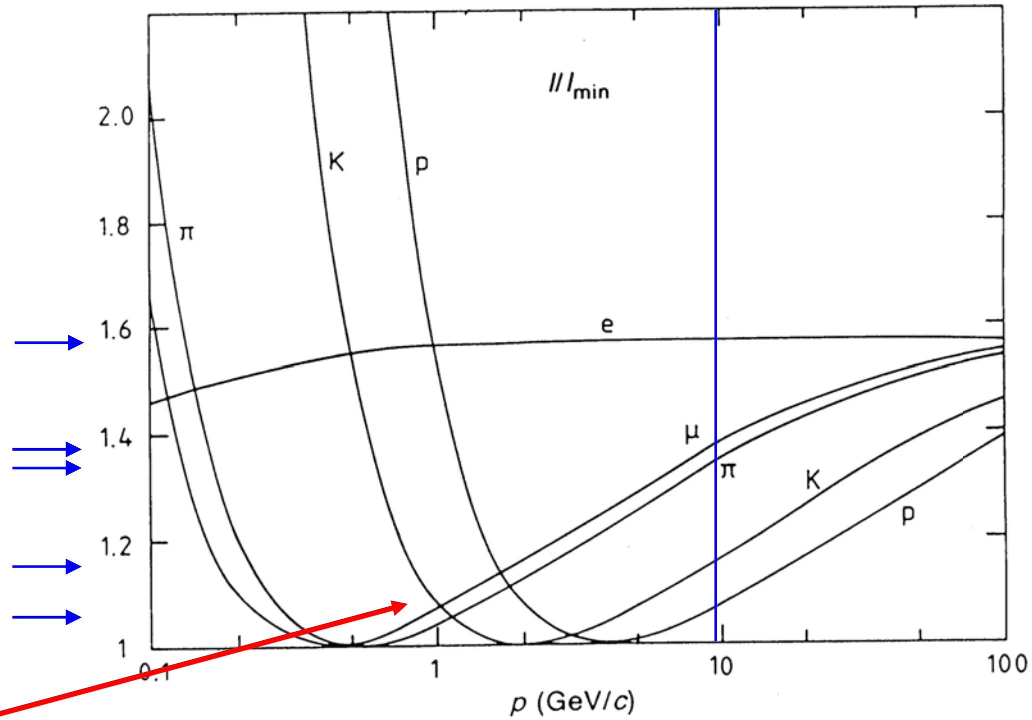
(We usually cheat and calculate E from p and m)

dE/dx

Charged particles moving through matter lose energy to ionization

Loss is a function of the speed, $\beta \equiv \frac{v}{c}$ **so a function of mass and momentum**

$$m = \frac{p}{\gamma\beta}$$



Alternately, measuring With certain ambiguities!

Its hard to make this precise

Minimize material -> small loses

- Hard to measure dE well

Geometry of tracking is complex

- Hard to measure dx well

Typical accuracy is 5-10%

- "2 sigma separation"

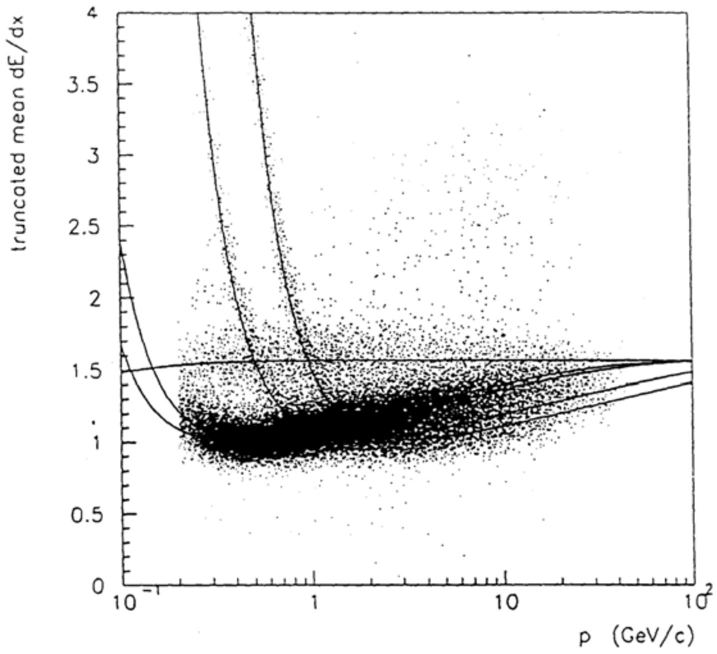


Fig. 8: Scatter plot of the ionisation measurement for a large set of hadronic Z_0 decays

During analysis, can choose

- **efficiency**
- **purity**

But can't have both!

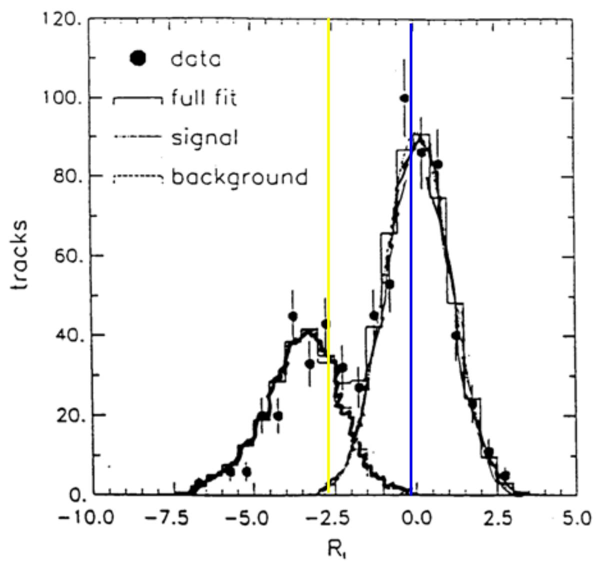
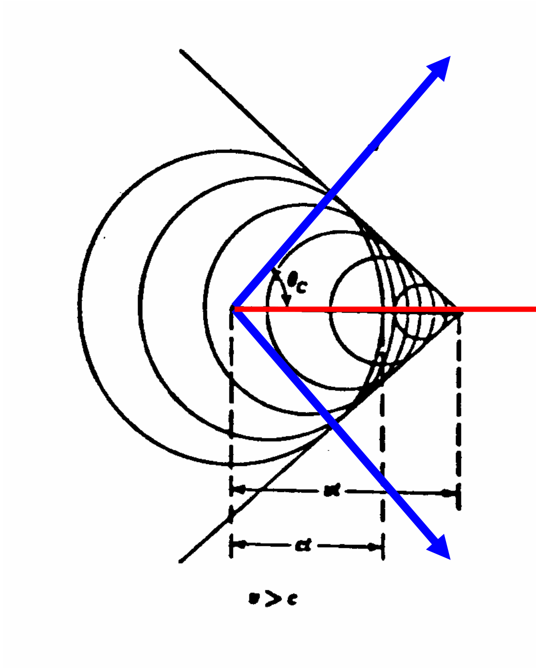


Fig. 10: Histogram of electron candidates using the dE/dx information of the TPC

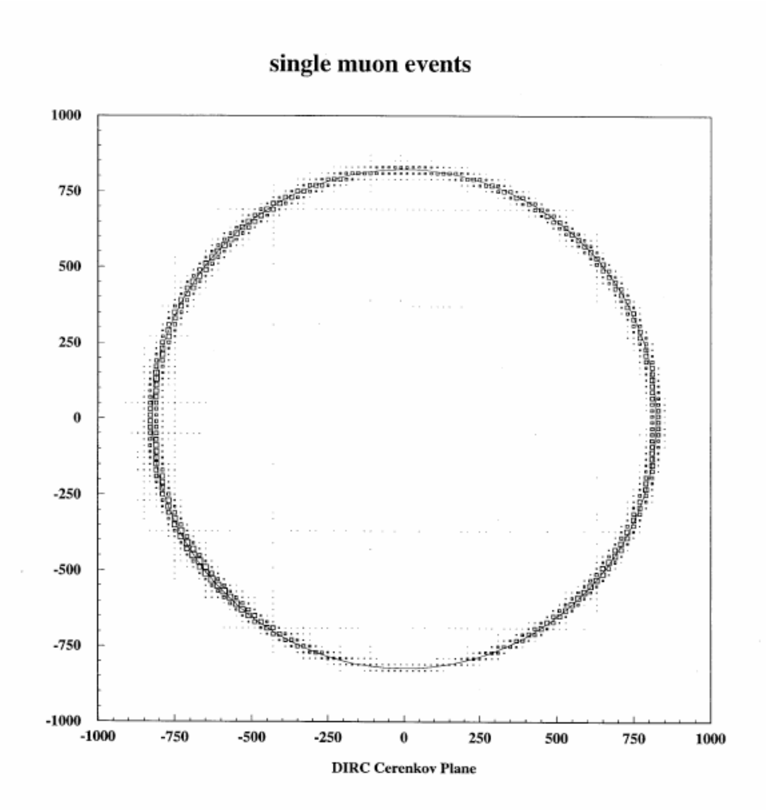
Another velocity-dependent process: Cherenkov light

Particles moving faster than light in a medium (glass, water) emit light

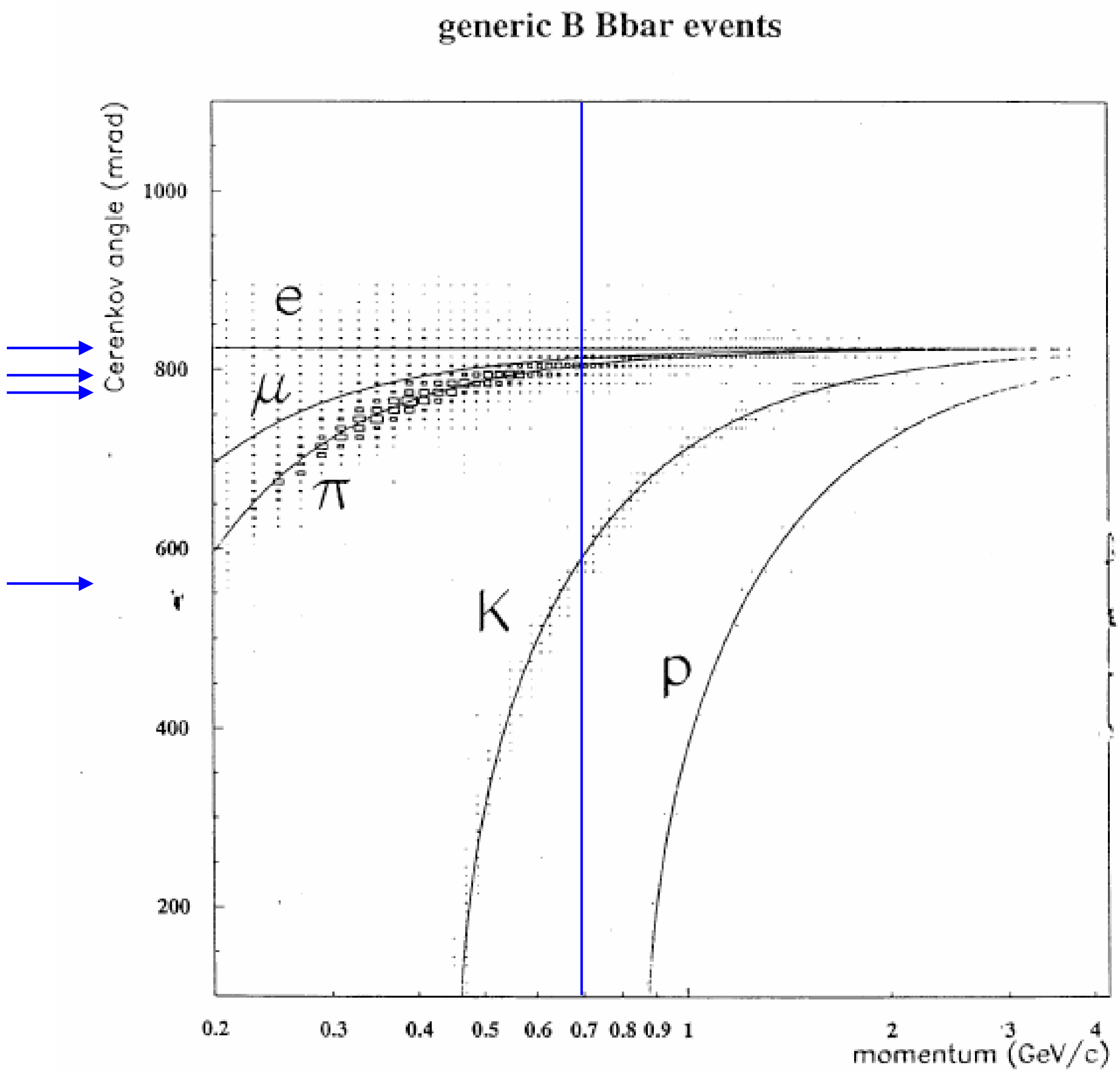
- Angle is related to velocity
- Light forms a cone



Focus it onto a plane, and you get a circle:



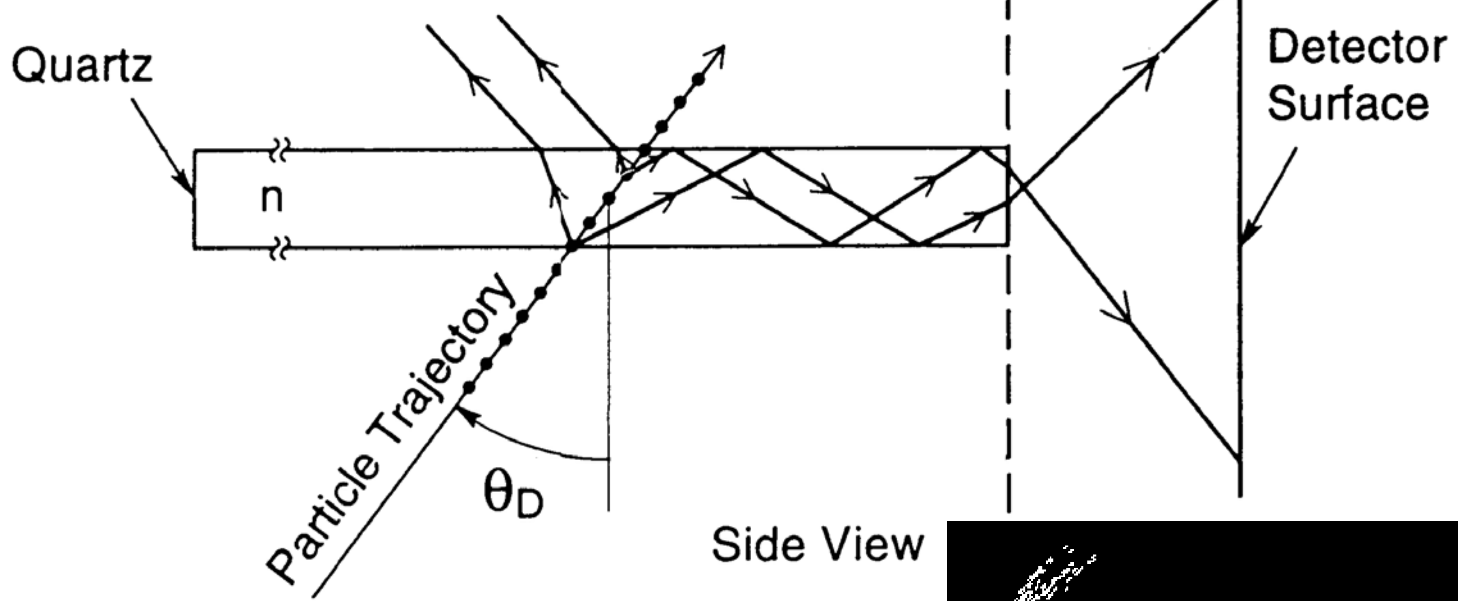
Radius of the reconstructed circle give particle type:

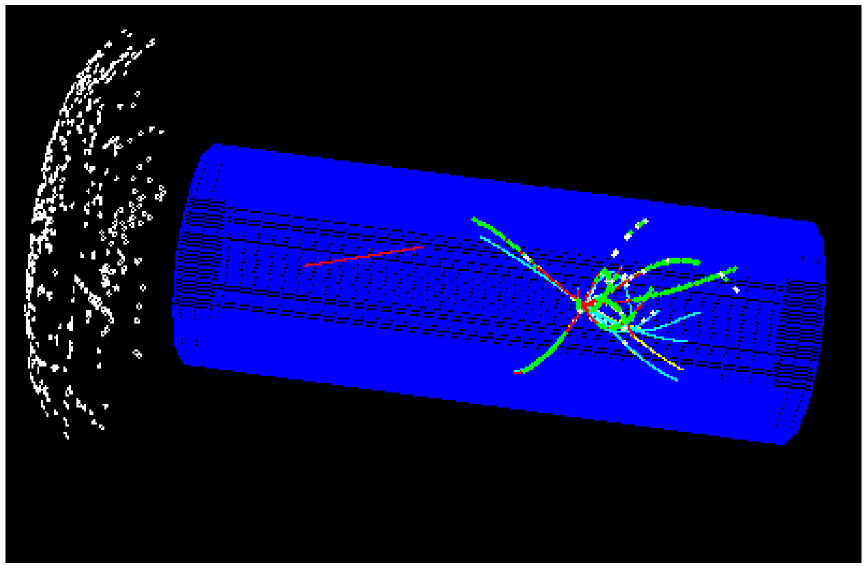


How to make this fit?

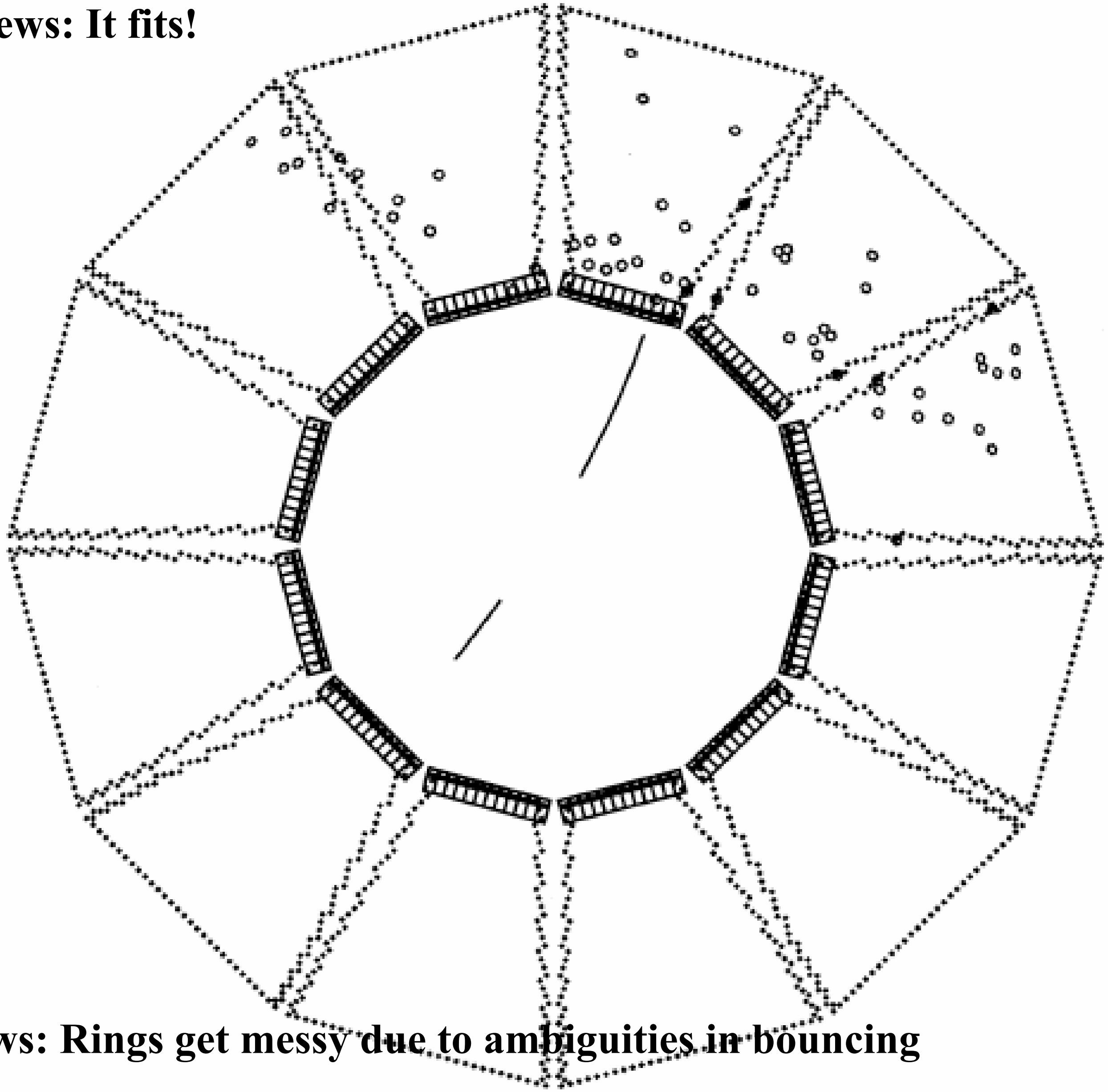
Space inside a detector is very tight, and the ring needs space to form

BaBar uses novel "DIRC" geometry:



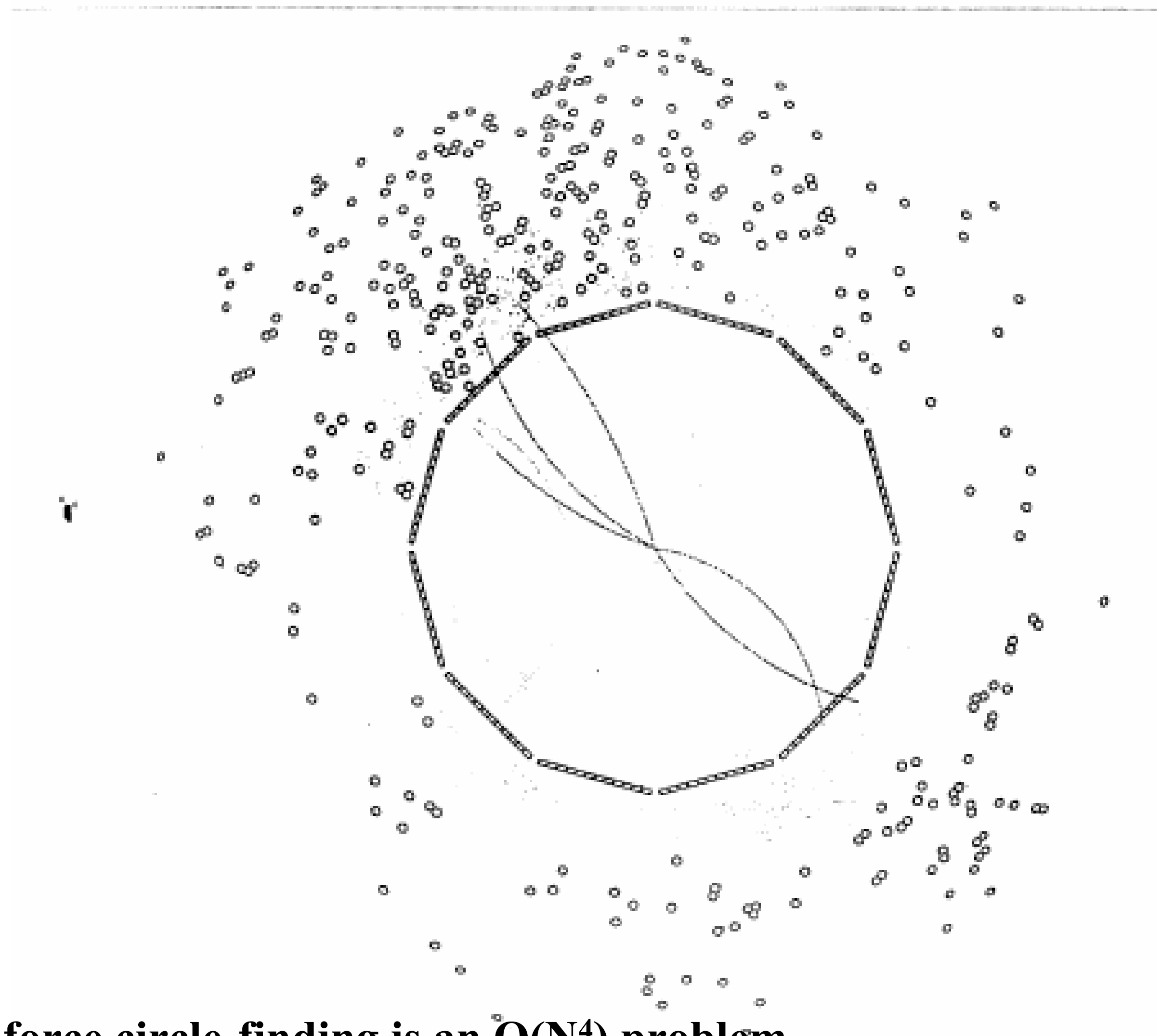


Good news: It fits!



Bad news: Rings get messy due to ambiguities in bouncing

Simple event with five charged particles:



Brute-force circle-finding is an $O(N^4)$ problem

Realistic solution?

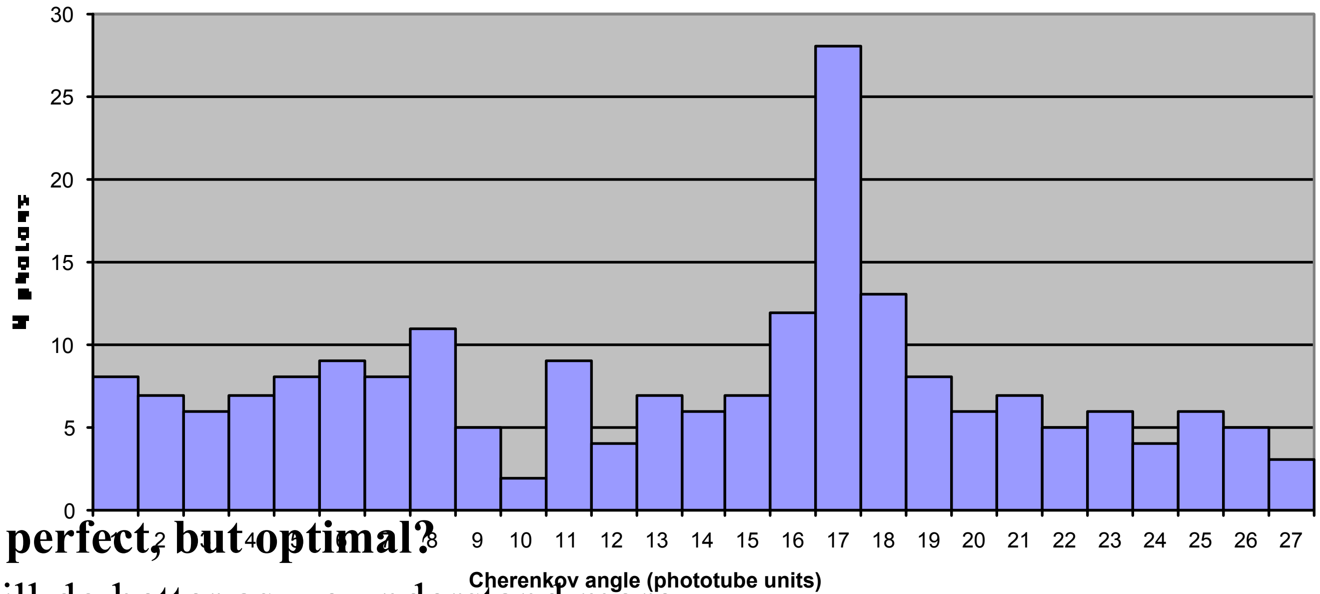
Use what you know:

- Have track trajectories, know position and angle in DIRC bars
- All photons from a single track will have the same angle w.r.t. track

  No reason to expect that for photons from other tracks

For each track, plot angle between track and <u>every</u> photon

- Don't do pattern recognition with individual photons
- Instead, look for overall pattern



Not perfect, but optimal?

Will do better as we understand more

What about the computing behind this?

BaBar records about 100k B events per day

- Hidden in 10 million events recorded/day
- Take data about 300 days per year

'Prompt processing'

- Want data available in several days
- Reconstruction takes about 3 CPU seconds/evt
- Processed multiple times
  - E.g. new algorithms, constants, etc

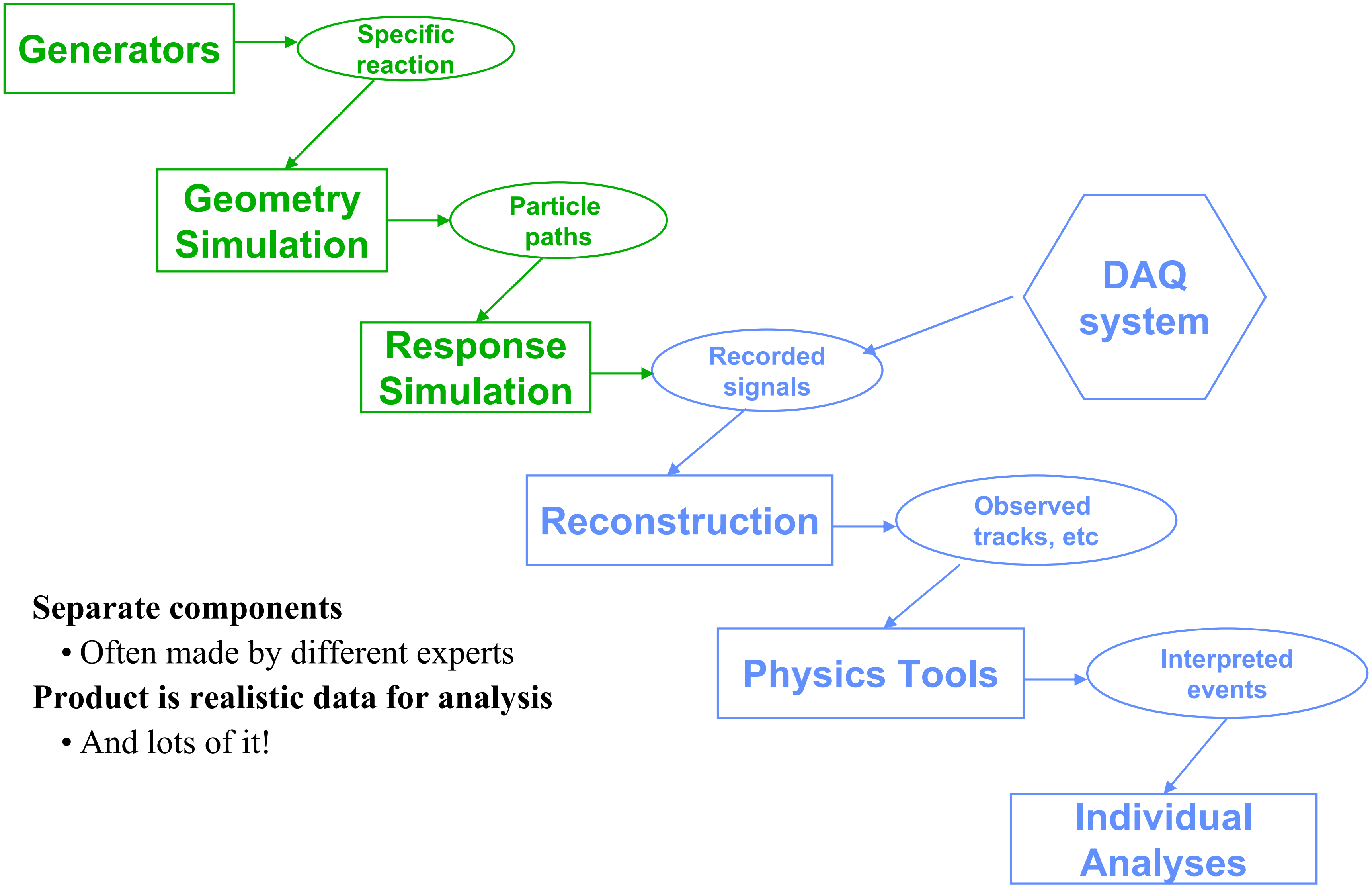
We have about 3000 million simulated events to study

- About half in specific decay modes
- Half 'generic' decays to all modes

About 4 million lines of code in simulation and reconstruction programs

- Plus the individual analyses

Traditional flow of data - real and simulated



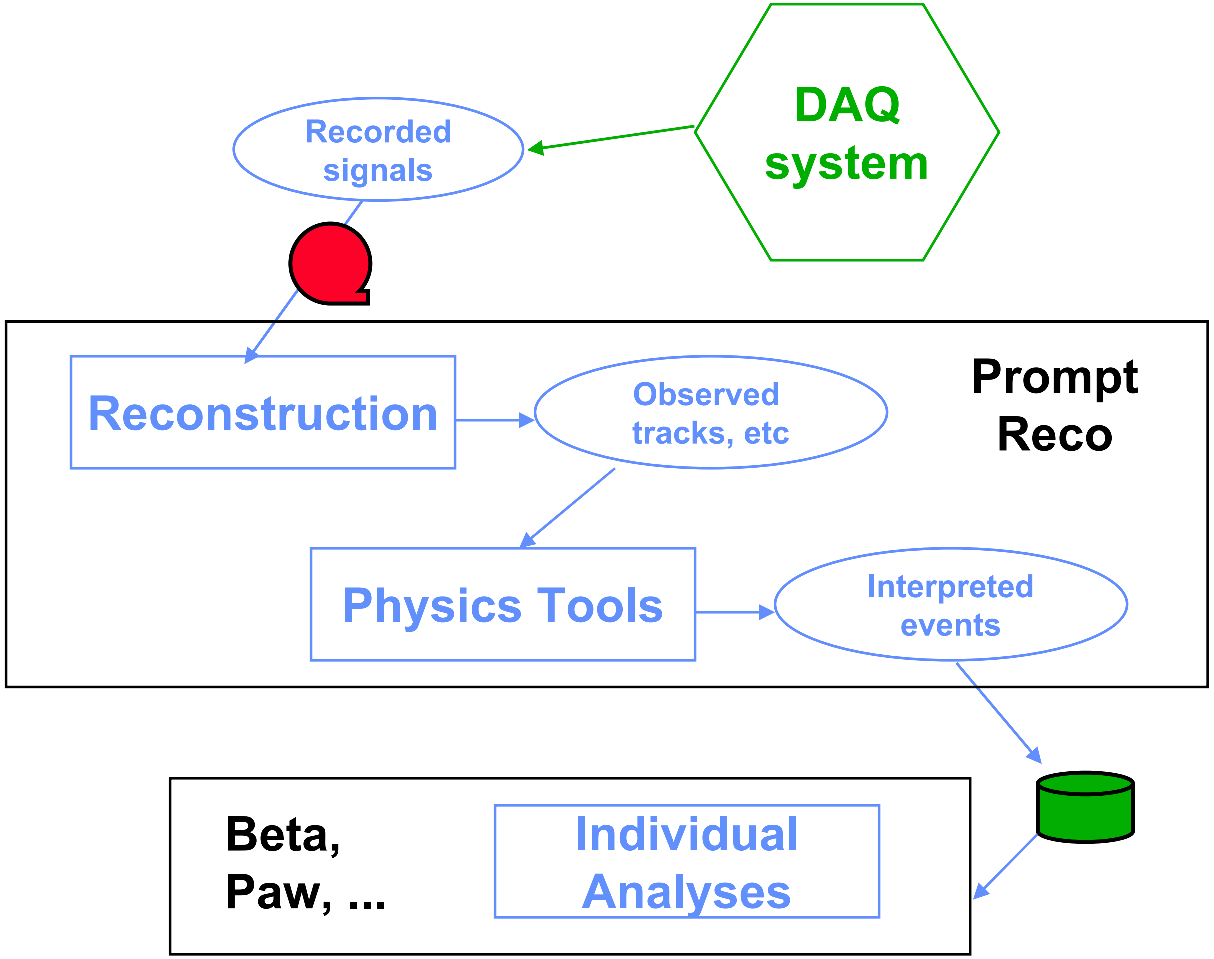
Separate components

- Often made by different experts

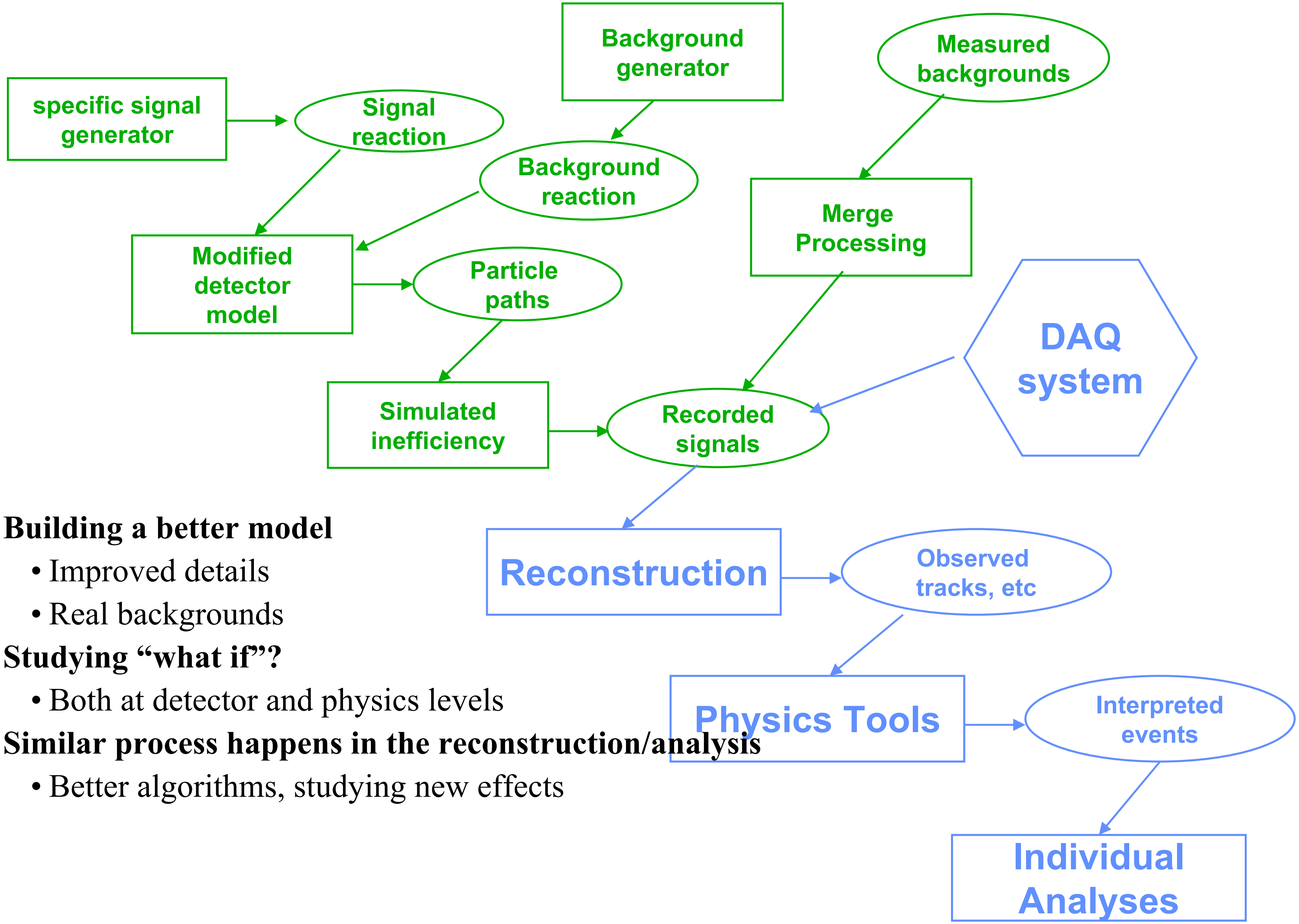
Product is realistic data for analysis

- And lots of it!

Processing real data



More detailed studies via more detailed simulation



Building a better model

- Improved details
- Real backgrounds

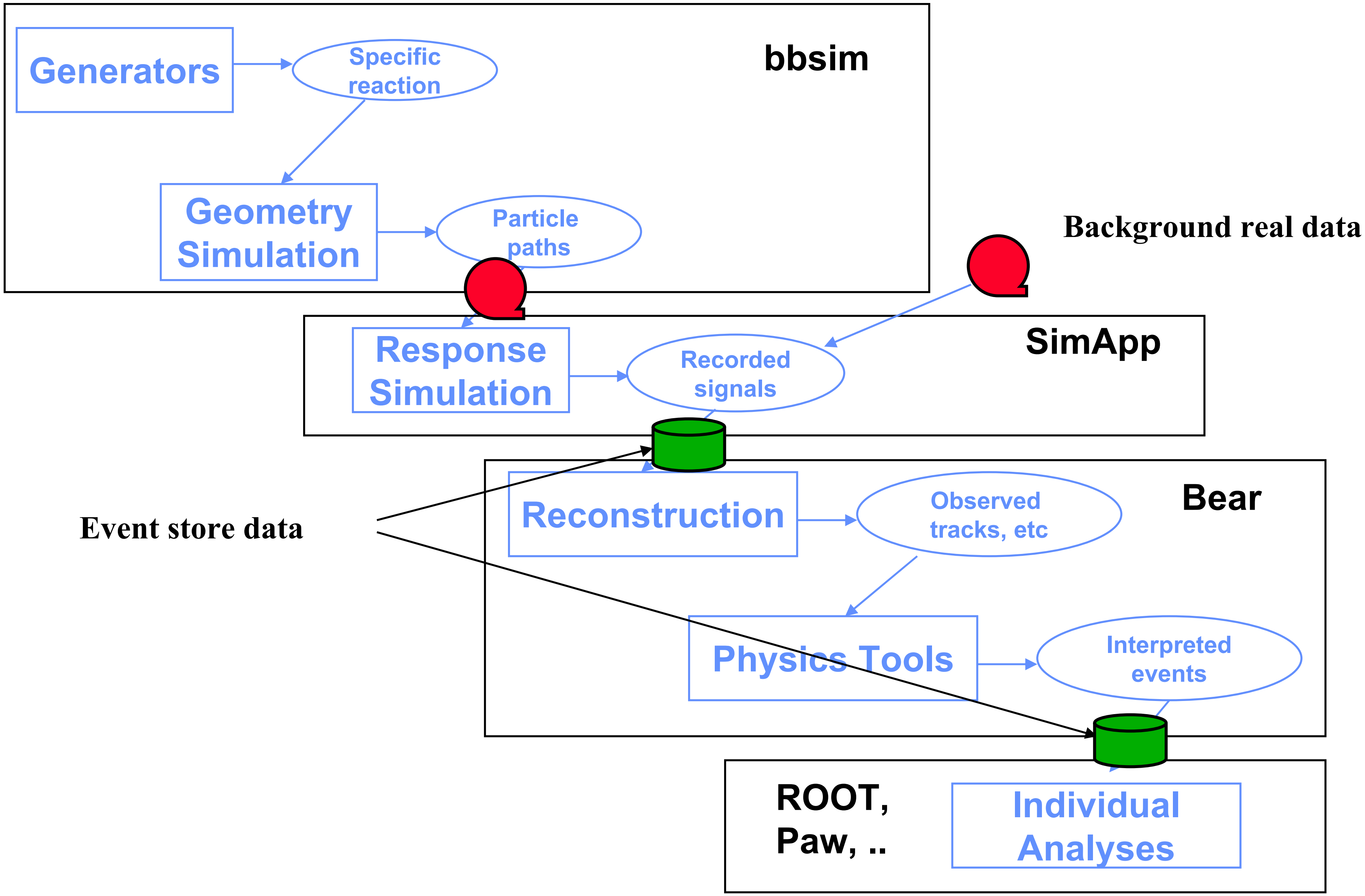
Studying "what if"?

- Both at detector and physics levels

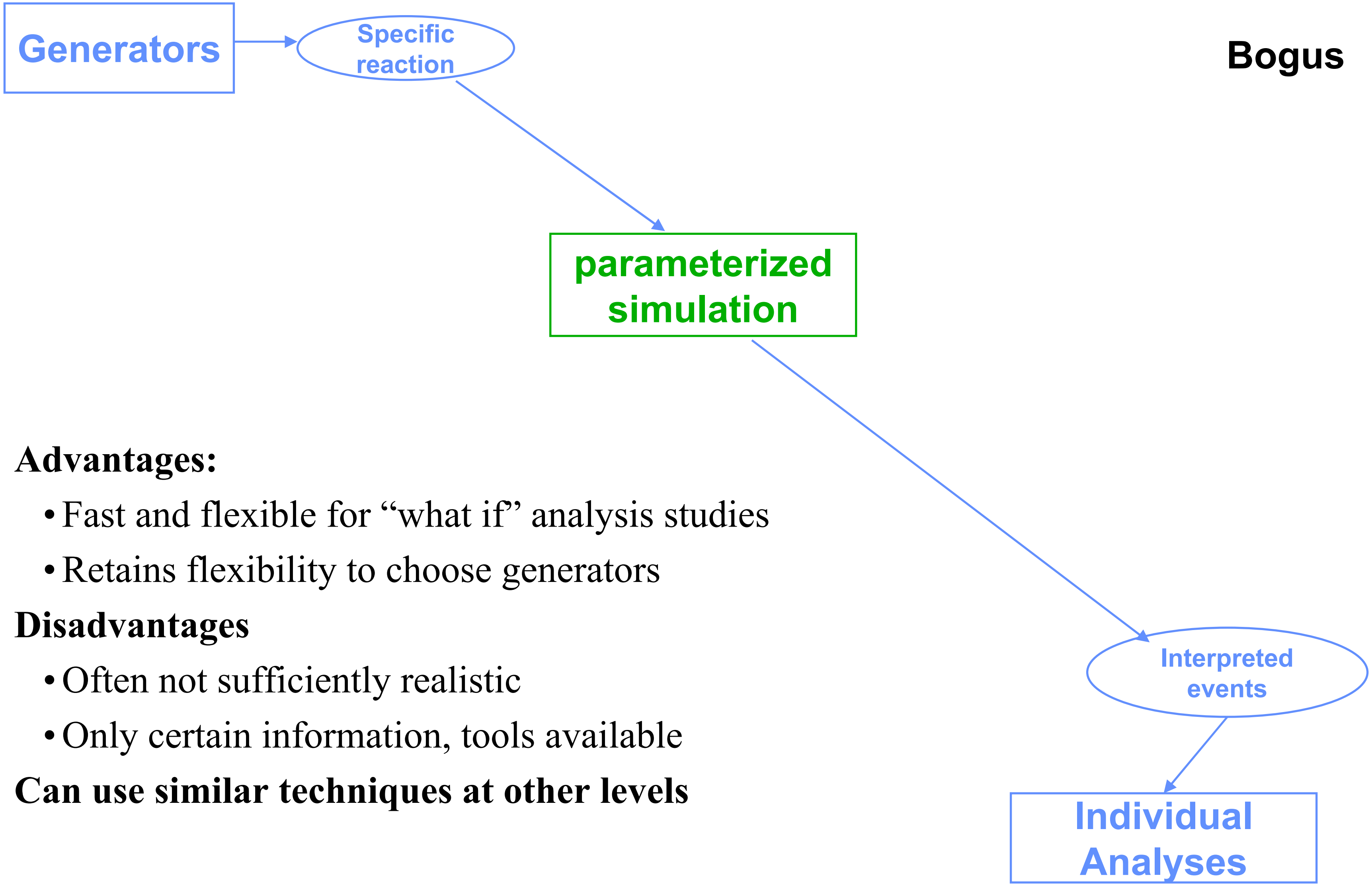
Similar process happens in the reconstruction/analysis

- Better algorithms, studying new effects

Partitioning production system into programs



Speed, simplify simulation by crossing levels



Advantages:

- Fast and flexible for "what if" analysis studies
- Retains flexibility to choose generators

Disadvantages

- Often not sufficiently realistic
- Only certain information, tools available

Can use similar techniques at other levels

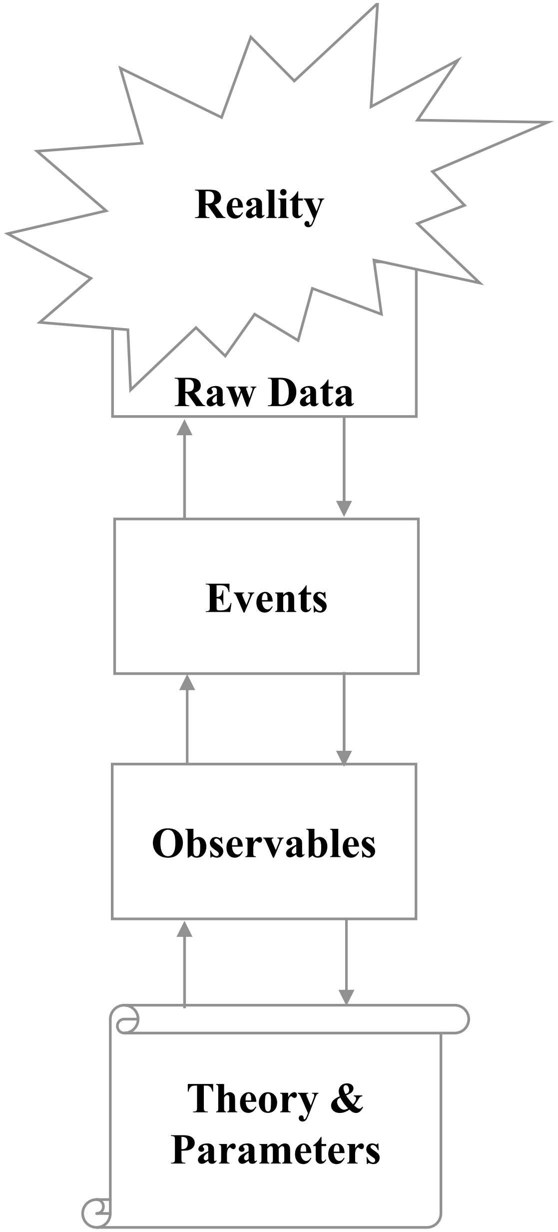
Why do we do this?

Detailed simulations are part of HEP physics

- Simulations are present from the beginning of an experiment
  - Simple estimates needed for making detector design choices
- We build them up over time
  - Adding/removing details as we go along
- We use them in many different ways
  - Detector performance studies
  - Providing efficiency, purity values for analysis
  - Looking for unexpected effects, backgrounds

Why do we use such a structure?

- Flexibility - we have different versions of the pieces
  - Comparison forms an important cross check
- Efficiency
  - We build up collections of data at each step for repeated study
  - "I found this background effect in the Spring dataset…"
- Manageability
  - Large programs are hard to build, understand, use



The imperfect measurement of a (set of) interactions in the detector

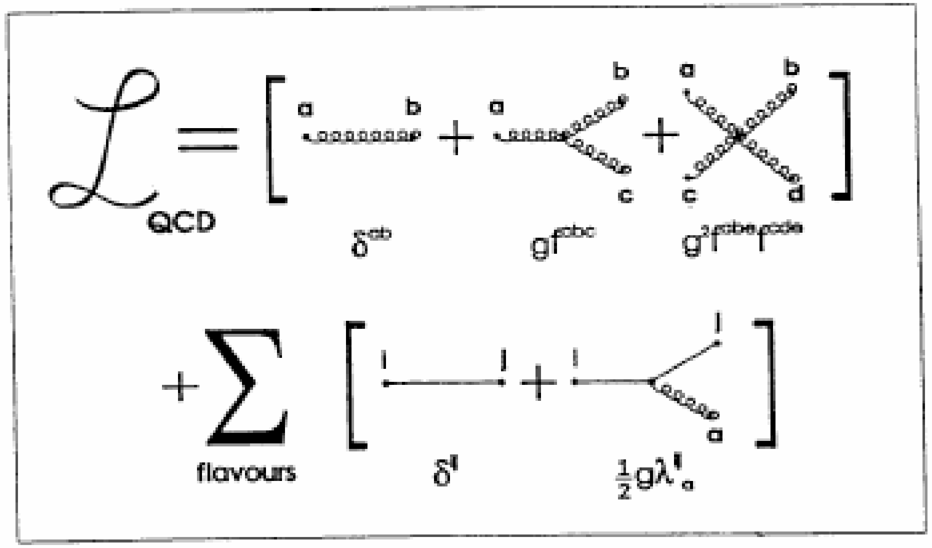
A unique happening:
Run 21007, event 3916 which contains a J/psi -> ee decay

Specific lifetimes, probabilities, masses, branching ratios, interactions, etc

A small number of general equations, with specific input parameters (perhaps poorly known)

Analysis: Measuring α_S in QCD

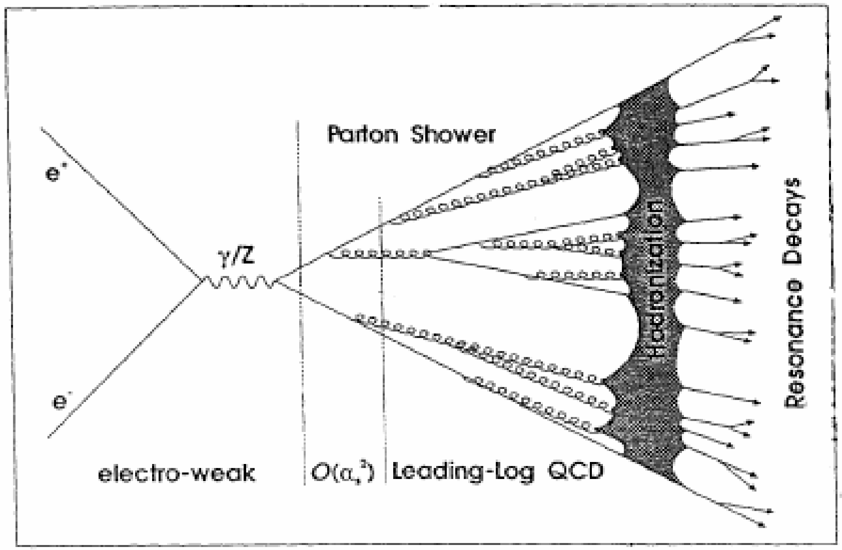
QCD predicts a set of basic interactions:



- You can measure the strong coupling constant by the relative rates

Unfortunately, QCD only makes exact predictions at high energy

- Low energy QCD, e.g. making hadrons, must be "modeled"



Compare models to observations in lots of different variables

Over time, new models get created and old ones improve

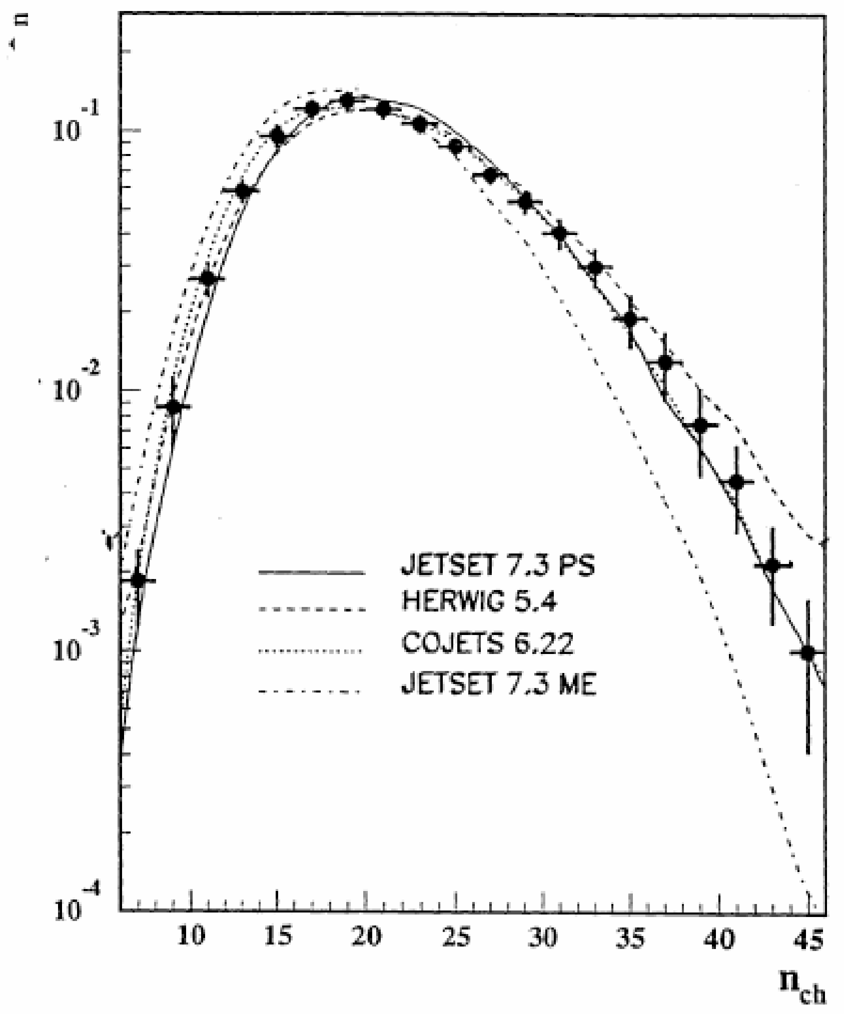
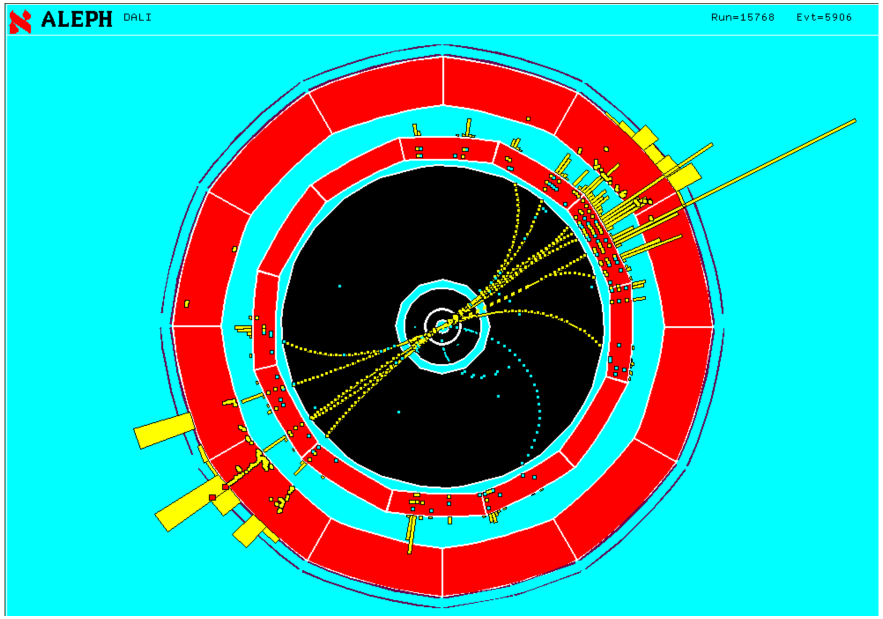
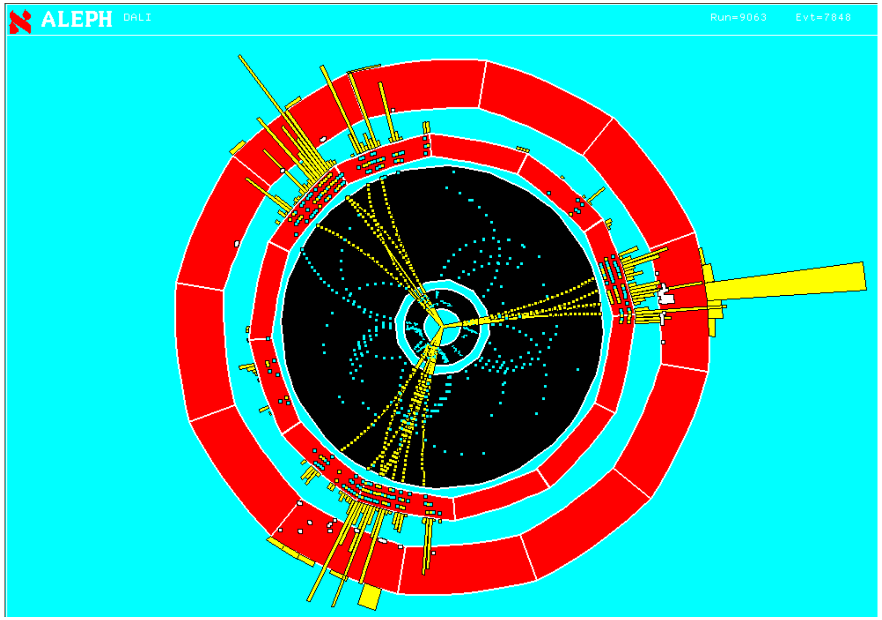


Figure 5: Charged multiplicity distribution measured by the L3 collaboration [28]. The points with error bars are the experimental data, the curves are model predictions.

"Jets"

Groups of particles probably come from the underlying quarks and gluons





But how to make this more quantitative?

- Don't want people "guessing" at whether there are two or three jets
- Need a jet-finding algorithm

Simple one:

- Take two particles with most similar momentum and combine into one
- Repeat, until you reach a stopping value "y_{cut}"

What about that arbitrary cut?

Nature doesn't know about it

- If your model is right, your simulation should reproduce the data at any value of the cut
- Pick one (e.g. 0.04), and use the number of 2,3,4, 5 jet events to determine α_S.
- Then check consistency at other values, with other models

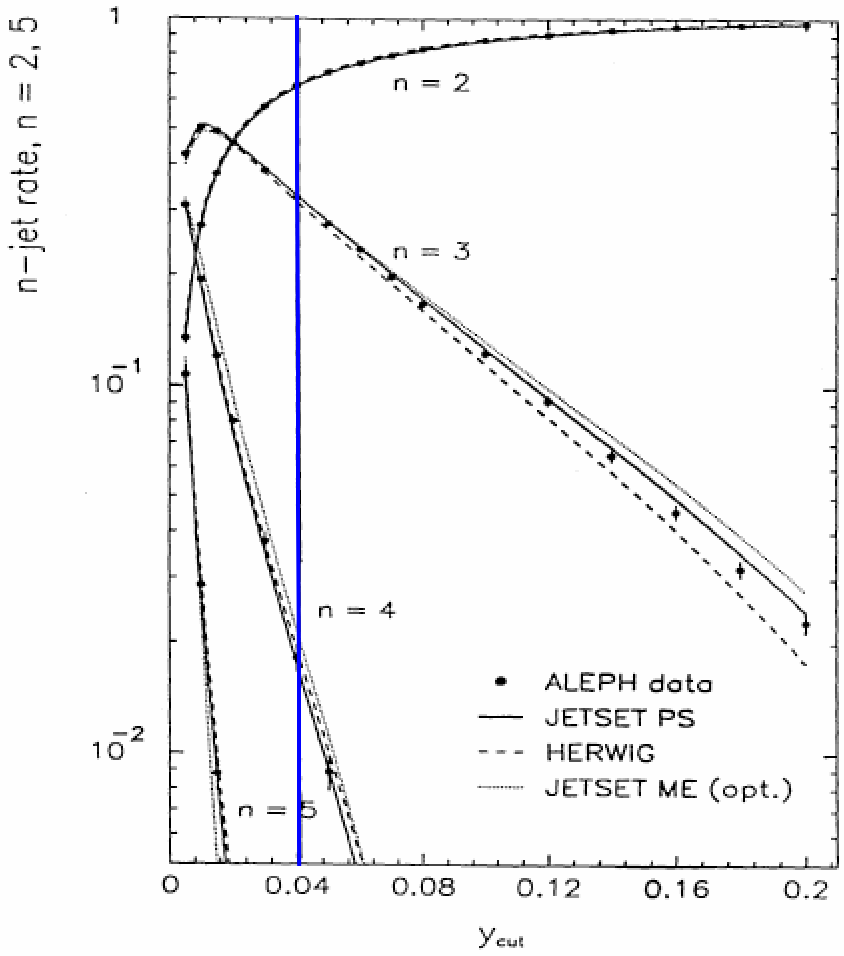


Figure 8: Jet rates determined by the ALEPH-collaboration [29] as function of the jet resolution parameter y_{cut}. The experimental results are compared to model calculations. Note that neighbouring points are highly correlated.

Many ways to measure α_s

If the theory's right, all get same value because all are measuring same thing

If the values are inconsistent, perhaps a more complicated theory is needed

Or maybe we just made a mistake...

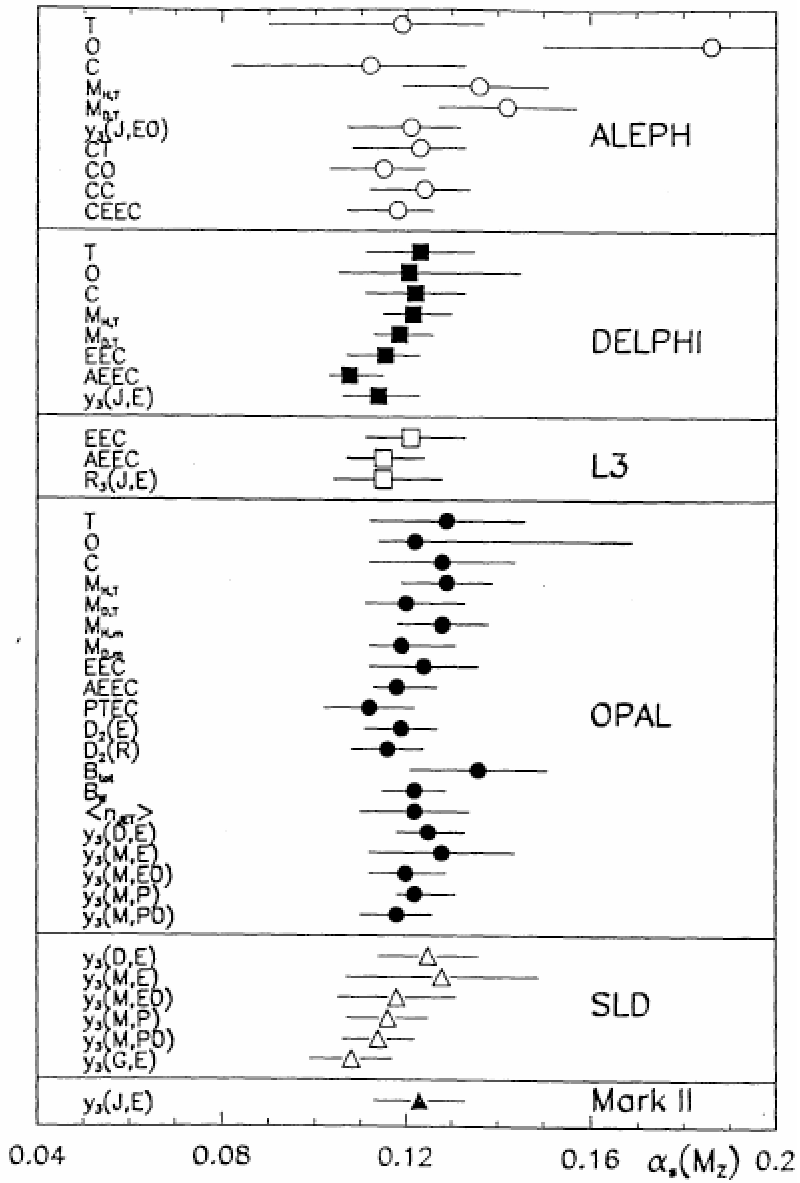


Figure 12: Measurements of the strong coupling constant from event shape variables based on second order QCD predictions.

Alignment & Calibration

How do you know the gain of each calorimeter cell?

- What's the relationship between ADC counts and energy?
- You designed it to have a specific value; does it?

How do you know where the tracking hits are in space?

- Need to know Si plane positions to about 5 microns

Start with

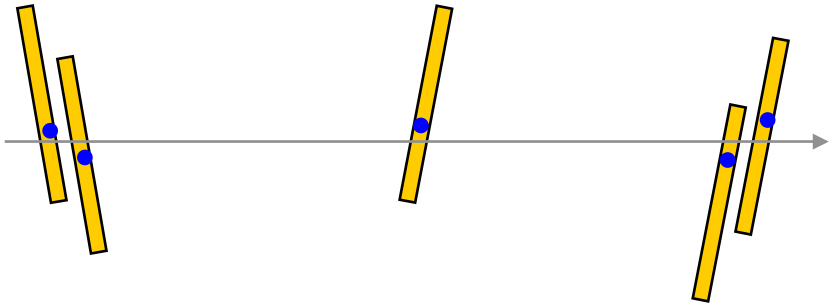
- Test beam information
- Surveys during construction
- Simulations and tests

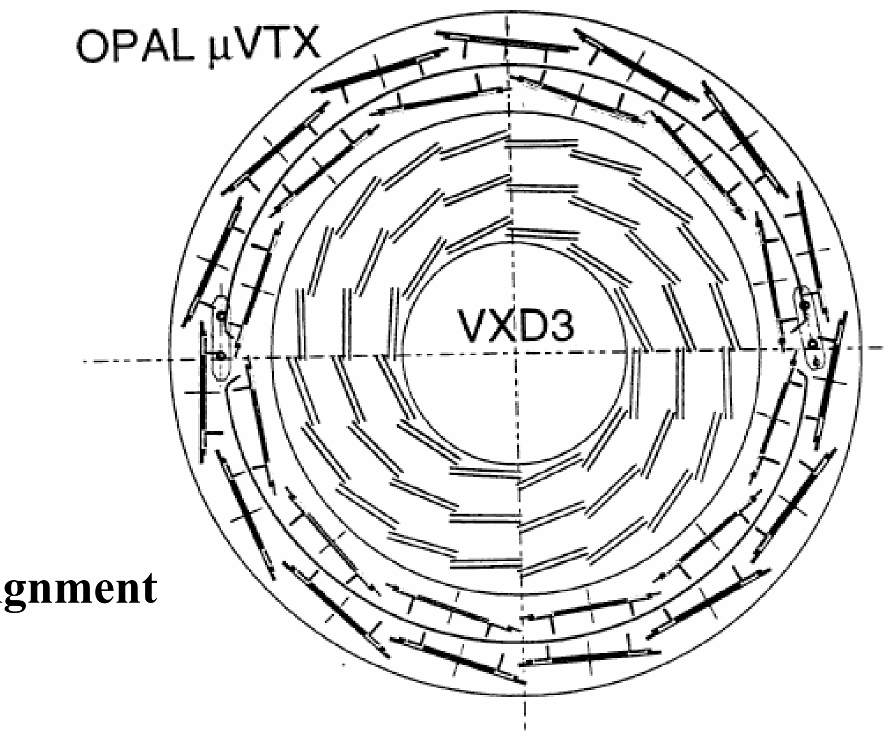
But it always comes down to calibrating/aligning with real data

Example: BaBar vertex detector alignment

About 700 Si wafers

- Each with 6 degrees of freedom
- => 4200 alignment constants to find





Small motions => small changes in alignment

=> change χ^2 of track

Approach 1: Take 10^5 tracks

Calculate sum of track χ^2s

For each of 4200 constants, generate equation from $$\frac{\partial \chi^2}{\partial c_i} = 0$$

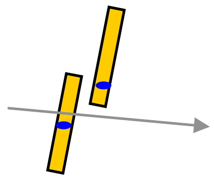
Solve 4200 equations in 4200 unknowns

Computationally infeasible

- Even worse, non-linear fit won't converge

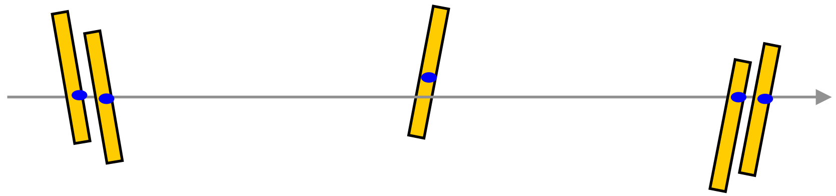
Instead, break problem into pieces:

- Two mechanical halves => 2x6 “global alignment constants”
- “local” constants within the halves



Do local alignment iteratively

- Look at pairs of adjacent wafers, and try to position them
- Then use tracks to position entire layers



- And iterate as needed

Iterative, sensitive process

- Manually guided from initial knowledge to final approximation
- Requires judgement on when to stop, how often to redo

Summary

Reconstruction and analysis is how we get from raw data to physics papers

Throughout, you deal with:

- Too little information
- Too much detail
- Little prior knowledge

You have to count on

- Lots of cross checks
- Prior art
- Tuning and evolutionary improvement

But you can generate wonderful results from these instruments!